

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 1月31日  
Date of Application:

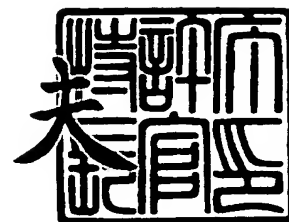
出願番号 特願2003-024861  
Application Number:  
[ST. 10/C]: [JP2003-024861]

出願人 株式会社東芝  
Applicant(s):

2003年 7月18日

特許庁長官  
Commissioner,  
Japan Patent Office

今井 康



【書類名】 特許願

【整理番号】 A000300321

【提出日】 平成15年 1月31日

【あて先】 特許庁長官 殿

【国際特許分類】 G11C 16/06

【発明の名称】 制御回路及びメモリコントローラ

【請求項の数】 12

【発明者】

【住所又は居所】 東京都青梅市末広町 2 丁目 9 番地 株式会社東芝青梅事業所内

【氏名】 鷹居 頼治

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

## 【選任した代理人】

【識別番号】 100108855

【弁理士】

【氏名又は名称】 蔵田 昌俊

## 【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

## 【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

## 【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 制御回路及びメモリコントローラ

【特許請求の範囲】

【請求項 1】 記憶装置から読み出したデータを全ビット反転する反転手段と、

前記反転手段の出力をエラー訂正復号処理する復号手段と、  
を具備する制御回路。

【請求項 2】 前記復号手段は記憶装置のデータ消去後の初期値の全ビット反転値に対してエラー無しを検出する請求項 1 に記載の制御回路。

【請求項 3】 前記記憶装置は不揮発性半導体記憶装置を具備する請求項 1 に記載の制御回路。

【請求項 4】 記憶装置に書き込まれるデータを全ビット反転する第 1 の反転手段と、

前記第 1 の反転手段の出力をエラー訂正符号化処理する符号化手段と、  
前記符号化手段から出力されるデータを全ビット反転して記憶装置に書き込む第 2 の反転手段と、

記憶装置から読み出したデータを全ビット反転する第 3 の反転手段と、  
前記第 3 の反転手段の出力をエラー訂正復号処理する復号手段と、  
を具備する制御回路。

【請求項 5】 前記符号化手段の符号化方式は、前記復号手段が記憶装置のデータ消去後の初期値の全ビット反転値に対してエラー無しを検出する方式である請求項 4 に記載の制御回路。

【請求項 6】 前記復号手段は記憶装置のデータ消去後の初期値の全ビット反転値に対してエラー無しを検出する請求項 4 に記載の制御回路。

【請求項 7】 前記記憶装置は不揮発性半導体記憶装置を具備する請求項 4 に記載の制御回路。

【請求項 8】 記憶装置に接続されるメモリコントローラにおいて、  
データを一時的に保持するバッファと、  
前記バッファから記憶装置に書き込まれるデータを全ビット反転する第 1 の反

転手段と、

前記第1の反転手段の出力をエラー訂正符号化処理する符号化手段と、

前記符号化手段から出力されるデータを全ビット反転して記憶装置に書き込む  
第2の反転手段と、

記憶装置から読み出したデータを全ビット反転する第3の反転手段と、

前記第3の反転手段の出力をエラー訂正復号処理する復号手段と、

を具備するメモリコントローラ。

【請求項9】 前記符号化手段の符号化方式は、前記復号手段が記憶装置のデータ消去後の初期値の全ビット反転値に対してエラー無しを検出する方式である請求項8に記載のメモリコントローラ。

【請求項10】 前記復号手段は記憶装置のデータ消去後の初期値の全ビット反転値に対してエラー無しを検出する請求項8に記載のメモリコントローラ。

【請求項11】 前記バッファの出力と、前記符号化手段の出力とを選択的に記憶装置に供給する手段をさらに具備する請求項8に記載のメモリコントローラ。

【請求項12】 前記記憶装置は不揮発性半導体記憶装置を具備する請求項8に記載のメモリコントローラ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は半導体記憶装置等のエラーチェック訂正 (Error Checking and Correction or Error Correction Code) (以下、ECCと略称する) に関する制御回路及びメモリコントローラに関する。

【0002】

【従来の技術】

不揮発性半導体記憶装置の一例としてフラッシュメモリがある。例えば、NAND型のフラッシュメモリでは、512バイト単位のブロック毎に書き込みが行われ、ブロックは不連続でも構わないが、書き込みアドレスは増加しながら書き込まれるものもある。そのため、不連続な複数のブロックに一旦データを書き込んだ

後、中間のブロック（アドレスが小さい）にデータを書込みたい場合は、一旦、他の領域へデータを移してから、アドレスが増加するブロックに順番に書き込む必要がある。この時、データが書き込まれていない領域からデータを読み出すことになるが、データが書き込まれていない領域は、データが消去されている消去領域である。不揮発性半導体記憶装置において、データの消去を行うと、その領域のデータはある初期値（例えば、すべてのビットが“1”）となる。

#### 【0003】

一方、不揮発性半導体記憶装置に書き込まれるデータはエラー訂正符号化処理され、本来の書き込みデータに対してECCビット（1ビット、あるいは数ビット）が付加されている。不揮発性半導体記憶装置から読み出したデータはエラー訂正復号処理される。消去領域のデータは例えば、ビット列がオール“1”であるが、通常のECC符号、例えば拡張ハミング符号では、ビット列がオール“1”、オール“0”のデータに対するECCビットはオール“1”、オール“0”にはならない。このため、消去領域から読み出したデータは誤まりは無い（正しい）にも係わらず、ECCエラー有りとは誤判定されてしまう。

#### 【0004】

このようなECCエラー誤判定を防ぐために、ECC制御回路内にエンコーダ用、デコーダ用の2つのインバータを内蔵し、ECCエンコーダの出力、ECCデコーダの出力の適宜ビットを反転することにより、ECCエラーが誤検出されないようにすることが考えられている（特許文献1参照）。

#### 【0005】

特許文献1に記載のECC制御回路は、チェックビット生成回路、シンドロームデコーダ（ECCデコーダ）、チェックビット生成回路から生成されたチェックビットの少なくとも一部を反転する第1のビットインバータ、チェックビット格納用不揮発性メモリから読み出したチェックビットの少なくとも一部を反転する第2のビットインバータから構成されている。

#### 【0006】

第1のビットインバータは、データ領域の初期値に対して生成されるチェックビットが消去後の初期値と等しくなるようにその一部を反転する。第2のビット

インバータは、チェックビット格納用不揮発性メモリから読み出したチェックビットについて第1のビットインバータと同じビットの反転を行う。

#### 【0007】

データ書込みの際は、データをデータ用不揮発性メモリへ書き込むとともにチェックビット生成回路へ供給する。チェックビット生成回路は、データに応じたチェックビットを生成する。第1のビットインバータは、このチェックビットを、ECCエラーが発生しないように適宜ビット反転させ、チェックビット格納用不揮発性メモリに書き込む。

#### 【0008】

データ読出しの際は、データをデータ用不揮発性メモリから読み出し、ECC制御回路に供給し、チェックビットをチェックビット格納用不揮発性メモリから読出し、第2のビットインバータを介して、その一部が反転されてシンドロームデコーダへ供給される。シンドロームデコーダはエラーがあれば適宜修正し、修正後のデータをデータバスへ送出する。

#### 【0009】

このように、消去領域から読み出したデータからチェックビットを生成し、この一部のビットをエラーが発生しないように適宜反転させることにより、ECCエラーの誤検出を防止している。

#### 【0010】

##### 【特許文献1】

特開2001-92723号公報（段落0013～段落0014、図3）

#### 【0011】

##### 【発明が解決しようとする課題】

特許文献1記載では、ECC符号の特性を考慮していないため、ビットインバータが反転するビットは固定ではないので、インバータが複雑になる。一般に、集積回路は回路記述をした後、シミュレーションテストが必要であるが、複雑なインバータはテスト項目数も多く、テストに時間がかかる欠点がある。

#### 【0012】

この発明の目的は、消去後の初期値に対してECCエラーを誤検出することがなく、動作テストも容易な制御回路及びメモリコントローラを提供することである。

#### 【0013】

##### 【課題を解決するための手段】

上記した課題を解決し目的を達成するために、本発明は以下に示す手段を用いている。

#### 【0014】

制御回路は、

記憶装置から読み出したデータを全ビット反転する第1の反転手段と、  
前記第1の反転手段の出力をエラー訂正復号処理する復号手段と、  
を具備する。

#### 【0015】

記憶装置に接続されるメモリコントローラにおいて、  
データを一時的に保持するバッファと、

前記バッファから記憶装置に書き込まれるデータを全ビット反転する第1の反転手段と、

前記第1の反転手段の出力をエラー訂正符号化処理する符号化手段と、

前記符号化手段から出力されるデータを全ビット反転して記憶装置に書き込む第2の反転手段と、

記憶装置から読み出したデータを全ビット反転する第3の反転手段と、

前記第3の反転手段の出力をエラー訂正復号処理する復号手段と、

を具備する。

#### 【0016】

この発明によれば、簡単な構成で、記憶装置の消去後の初期値に対してECCエラーを誤検出することがなく、動作テストも容易である。

#### 【0017】

##### 【発明の実施の形態】

以下、図面を参照して本発明による制御回路及びメモリコントローラの実施の



形態を説明する。ここでは、説明の便宜上、不揮発性半導体記憶装置はフラッシュメモリであるとし、消去後の初期値のビット列はオール“1”であるとし、誤り訂正符号をリードソロモン符号とする。図1はリードソロモン符号化回路への入力データとその出力符号との関係を表す。リードソロモン符号は入力データのEXORの組み合わせで構成されているため、データのビットがオール“0”の場合は、リードソロモン符号もオール“0”になるが、データのビットがオール“1”の場合は、リードソロモン符号はほとんどの場合オール“1”にはならない。このため、消去後の初期値がオール“1”ということは、データビットと符号ビットがともにオール“1”ということであり、ECCエラー有りと判定されてしまう。

#### 【0018】

##### 第1の実施の形態

図2は本発明の第1の実施の形態に係る制御回路を含むシステムの全体構成を示す図である。本実施の形態は、消去後の初期値からECCエラーが検出されないようにデータバスやECC符号バスにインバータを接続するものである。

#### 【0019】

フラッシュメモリ22にデータバス、制御バス、アドレスバスを介してメモリコントローラ24が接続される。メモリコントローラ24はデータバッファ26、マルチプレクサ28、DMAコントローラ30、ECCコントローラ32からなる。データバッファ26はフラッシュメモリ22へ書き込むデータ、フラッシュメモリ22から読み出すデータを一時的に保持する。ECCコントローラ32はフラッシュメモリ22へ書き込むデータにECC符号化処理、フラッシュメモリ22から読み出すデータにECC復号処理を施す。マルチプレクサ28はデータバッファ26からのデータとECCコントローラ32からのデータを選択的にフラッシュメモリ22へ供給する。DMAコントローラ30は、フラッシュメモリ22、データバッファ26にアドレスデータ、制御データを供給するとともに、ECCデコーダ34、ECCエンコーダ36に命令を供給する。

#### 【0020】

ECCコントローラ32は、データバッファ26から供給されるデータに基づ

いてECC符号を作成するECCエンコーダ36、メモリから読み出したデータとECC符号とに基づいてデータのエラーがあるか否かを判定し、エラーが訂正できる場合は訂正するECCデコーダ34、ECCデコーダ34へ供給されるデータ（メモリから読み出したデータ）の全ビットを反転するインバータ38、ECCデコーダ34へ供給されるECC符号（メモリから読み出したECC符号）の全ビットを反転するインバータ40、ECCエンコーダ36へ供給されるデータ（フラッシュメモリ22に書き込まれるデータ）の全ビットを反転するインバータ42、ECCエンコーダ36から出力されるECC符号の全ビットを反転するインバータ44からなる。

#### 【0021】

上記構成のシステムの動作を説明する。

#### 【0022】

##### データ書き込み

フラッシュメモリ22に書き込まれるデータがデータバッファ26に一旦保持される。DMAコントローラ30がデータバッファ26に対してデータバッファ用命令線62にリード命令を、データバッファ用アドレス線64に所望のアドレスを発行することにより、データバッファ26からデータ（メモリにライトするデータ）をリードする。DMAコントローラ30がフラッシュメモリ22に対してフラッシュメモリ用命令線78にライト命令を、フラッシュメモリ用アドレス線80に所望のアドレスを発行することにより、マルチプレクサ28を介してデータをフラッシュメモリ22に書き込む。DMAコントローラ30はECCエンコーダ36に対してエンコード命令線82にエンコード開始命令を送り、データバッファ26－フラッシュメモリ22間データバスからスヌープされたデータ74をインバータ42で反転し、反転データをECCエンコーダ36へ送る。データバッファ26からフラッシュメモリ22へデータを転送している際は、マルチプレクサ28はデータを通過させる処理を行う。マルチプレクサ28の出力であるフラッシュメモリ入力データバスを通してデータを転送し終わった後、ECCエンコーダ36はデータバッファ26から読み出したデータの反転値に対してECC符号を生成し、データ反転機能付きマルチプレクサ入力バス76を通じてE

CC符号の反転値を送信する。マルチプレクサ28はECC符号を通過させる処理を行い、このECC符号の反転値がフラッシュメモリ22へ書き込まれる。

#### 【0023】

上記手順に従いフラッシュメモリ22へオール“1”のデータを書き込むと、ECCエンコーダ36に入力されるデータは反転してオール“0”となるため、ECCエンコーダ36ではオール“0”のECC符号が生成される。これを反転してフラッシュメモリ22に書き込むため、データ、ECC符号ともにオール“1”が書き込まれる。

#### 【0024】

データ読み出し

DMAコントローラ30がフラッシュメモリ22に対してデータバッファ用命令線78にリード命令を、フラッシュメモリ用アドレス線80に所望のアドレスを発行することにより、フラッシュメモリ22からデータをリードする。DMAコントローラ30がデータバッファ26に対してデータバッファ用命令線62にライト命令を、データバッファ用アドレス線64に所望のアドレスを発行することにより、フラッシュメモリ22から読み出したデータをフラッシュメモリーデータバッファ間データバスを通してデータバッファ26に書き込む。DMAコントローラ30はECCデコーダ34に対してデコード命令線86を通じてデコード開始命令を送り、インバータ38の出力であるECCデコーダ入力用データ反転バスからリードデータの反転値を、インバータ40の出力であるECCデータ入力用ECC符号反転バスからECC符号の反転値をECCデコーダ34へ送る。

#### 【0025】

ECCデコーダ34のデコードの結果、データにエラーがあり修正が必要であった場合は、データ訂正動作が開始する。ECCデコーダ34が訂正箇所を特定・訂正する場合は、デコードステータス線84をビジー状態とし、訂正箇所特定・訂正中であることをDMAコントローラ30に通知し、データバッファ26へのリード・ライト動作を禁止させる。次に、ECCデコーダ34はデータバッファ26から訂正箇所の情報をECC訂正用リードデータバス66を通してリード

する。ECCデコーダ34は訂正すべきビット位置のエラーデータをビット反転により訂正し、訂正後のデータをECC訂正用ライトデータバス68を通してデータバッファ26にライトする。

#### 【0026】

上記手順に従いフラッシュメモリ22からオール“1”のデータを読み出すと、ECCデコーダ34に入力されるデータは反転してオール“0”となり、ECC符号も反転してオール“0”（上記書き込み手順を参照）であるため、図1に示すように、ECCエラーが検出されることがない。

#### 【0027】

以上説明したように、本実施の形態によれば、フラッシュメモリ22へ書き込む場合、フラッシュメモリ22にはECC符号として「データの反転値から計算されたリードソロモン符号の反転値」が書き込まれる。つまり、データ領域のビットがオール“1”の場合は、オール“1”のECC符号が書き込まれる。フラッシュメモリ22から消去後の初期値（オール“1”）を読み出す場合、ECCデコーダ34に入力されるデータはオール“0”、ECC符号もオール“0”となるため、ECCエラーが検出されることがない。

#### 【0028】

メモリ消去後の状態に対して、通常通りECC判定を行うとエラーが誤検出されてしまう。特許文献1ではメモリ消去後のデータからECC符号を生成し、この符号データをエラーが発生しないように適宜反転させて対応しているのに対し、本実施の形態ではメモリ消去後のデータの全ビットを反転させた後、ECC制御回路に入力し、ECC符号を生成している。特許文献1では、ECCエンコーダ・デコーダ内にインバータが実装されていることになるため、ECCエンコーダ・デコーダの単体テストが難しいが、本実施の形態では既存のECCエンコーダ・デコーダの外部にインバータを接続しているため、ECCエンコーダ・デコーダの単体テストが容易である。

#### 【0029】

以下、本発明による制御回路の他の実施の形態を説明する。他の実施の形態の説明において第1の実施の形態と同一部分は同一参照数字を付してその詳細な説

明は省略する。

#### 【0030】

##### 第2の実施の形態

図2に示す第1の実施の形態では、ECC制御はDMAコントローラ30が司っているが、DMAコントローラ30の代わりにCPUを用いて制御を行うこともできる。図3はこれを実現する第2の実施の形態のブロック図である。すなわち、DMAコントローラ30の代わりにCPU50が設けられる。CPU50を動作させるためのプログラムを格納するROMモジュール52、CPU50の作業領域となるRAMモジュール54が必要となる。DMAコントローラがECC制御を行う場合と比較して異なる制御を行うのは以下の2点である。

#### 【0031】

(1) ECC制御を行う前に、CPU50はプログラムリード用データ線92を通じてROMモジュール52からプログラムをロードする。

#### 【0032】

(2) CPU50はECC制御を行う際に、RAMアクセス用データ線94を通じて作業領域としてRAMモジュール54を使用する。

#### 【0033】

その他の制御は、第1の実施の形態と同様である。

#### 【0034】

以上説明したように、第2の実施の形態によっても、第1の実施の形態と同じ効果が得られる。

#### 【0035】

次に、上述した構成の第1、第2の実施の形態に適用されるLSIのテストを説明する。

#### 【0036】

図4は一般的なLSI設計フローを示す。ステップS10で、チップやパッケージ、コスト、周波数、回路規模見積もり等のLSI外部仕様を決定する。ステップS12で、外部仕様に従って、LSI内部の各モジュールの担当機能（内部仕様）を決定する。ステップS14で、内部仕様に従い、ハードウェア記述言語

を用いて回路を記述する。ステップ S 16 で、記述された回路の機能が正常に動作しているかどうかを検証する（機能シミュレーション）。ステップ S 18 で、シミュレーションが成功（回路機能が正常に動作している）か否かを判定する。

#### 【0037】

失敗の場合は、回路記述（ステップ S 14）に戻る。成功した場合は、ステップ S 20 で、ハードウェア記述言語で記述された回路を AND ゲートや OR ゲート等の論理ゲートに変換する（論理合成）。論理ゲートに変換された回路を以下「ネット」と称する。

#### 【0038】

ステップ S 22 で、ネットがハードウェア記述言語で記述された回路と同等の機能を持っているかどうかを検証する（論理シミュレーション）。検証内容は機能シミュレーションの検証テストと似通っている事が多い。ステップ S 24 で、ネットが希望動作周波数で正常に動作するかどうかを検証する（タイミング解析）。なお、ステップ 22 と 24 は並列で作業ができるため、どちらを先に実行しても良い。ステップ S 26 で、論理シミュレーションとタイミング解析が成功（ネットが正常に動作している）か否かを判定する。

#### 【0039】

機能シミュレーションにおいて正常な動作をしなかった場合、あるいはタイミング解析が成功しなかった場合には、回路記述を変更し、機能シミュレーションをやり直す（ステップ S 14 に戻る）。また、論理シミュレーションやタイミング解析でネットに異常が発見された場合は、論理合成をやり直したり（ステップ S 20 に戻る）、回路記述までさかのぼってやり直す（ステップ S 14 に戻る）。成功した場合は、ステップ S 28 で、実際に L S I 基板上に論理ゲートを配置、配線する（配置配線）。

#### 【0040】

図 4 の機能シミュレーション（ステップ S 16、S 18）の詳細を図 5 に示す。図中の単体機能とは ECC デコーダ 34、ECC エンコーダ 36、DMA コントローラ 30 等の単体のモジュールの機能を指す。これに対して、複合機能とは ECC デコーダ 34 とインバータ 38、40、ECC エンコーダ 36 とインバー

タ 42、44、ECCデコーダ34とECCエンコーダ36とインバータ38、40、42、44等のように複数のモジュールが組み合わさった機能を指す。ほとんどの場合、単体機能に比べて複合機能は機能が複雑になるため、その機能を検証するテストは多岐に渡り、テストパターンの作成や検証方法が難しくなることが多い。このためハードウェア言語で記述された回路を検証するためには、図5のように先ず単体機能テストを行って、単体の機能が正しいことが検証されてから、複合機能テストを行うのが一般的である。

#### 【0041】

例としてECCデコーダ34を挙げると、ECCデコーダ34の機能を検証するためには、ステップS52で、単体機能のためのテストパターンを作成する。ステップS54で、単体機能のシミュレーションを行う。ステップS56で、シミュレーションが成功か失敗か、失敗の場合、その原因がテストパターンにあるのか、回路記述にあるのかを判定し、原因に該当するステップに戻り、やり直す。

#### 【0042】

ステップS58で、複合機能（例えば、ECCデコーダとインバータを組み合わせた機能）テストのためのテストパターンを作成する。ステップS60で、複合機能のシミュレーションを行う。ステップS62で、シミュレーションが成功か失敗か、失敗の場合、その原因がテストパターンにあるのか、回路記述にあるのかを判定し、原因に該当するステップに戻り、やり直す。複合機能テストでは単体機能テストと異なり、それぞれのモジュールの機能の組み合わせの数（複合機能数）だけテストパターンが存在する。このテストパターンの数が多いと、それだけテスト作成に時間がかかるとともに、そのテストのシミュレーション時間が増大するため、LSI開発が現実的には困難となっている。

#### 【0043】

上述した本実施の形態のデータ、ECC符号を反転させるインバータは、リードソロモン符号の特性を利用して一意に全ビットを反転させているため、機能シミュレーションのためのテストパターンは1パターンである。しかし、特許文献1のインバータは、データ用フラッシュメモリやチェックビット用フラッシュメ

モリの特性に合わせて適宜チェックビットを反転させる必要があり、そのテストパターンは2のチェックビット乗必要となる。このため、複合機能テストパターンの作成や複合機能シミュレーションを2のチェックビット乗回数だけ実施する必要があり、全テストパターンだけ機能シミュレーションを実施することは非常に困難である。論理シミュレーションもこのテストパターンを使用するので、論理シミュレーションの実施も困難である。

#### 【0044】

本発明は上述した実施の形態に限定されず、種々変形して実施可能である。例えば、上述の説明では、ECC符号としてはリードソロモン符号を説明したが、他のECC符号を用いてもよい。例えば、オール“1”の時にECCエラーが必ずオール“0”のときにECCエラーになるECC符号化方式と消去後オール“0”になるメモリでも実装可能である。また、メモリの消去後の初期値はオール“1”としたが、これに限られない。

#### 【0045】

なお、本願発明は上記実施の形態に限定されるものではなく、実施段階ではその趣旨を逸脱しない範囲で種々に変形することが可能である。また、各実施の形態は可能な限り適宜組み合わせで実施してもよく、その場合組合わせた効果が得られる。さらに、上記実施の形態には種々の段階の発明が含まれており、開示される複数の構成要件における適宜な組み合わせにより種々の発明が抽出され得る。例えば、実施の形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題が解決でき、発明の効果の欄で述べられている効果が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

#### 【0046】

##### 【発明の効果】

以上説明したように本発明によれば、記憶装置の消去後の初期値に対してECCエラーを誤検出することがなく、動作テストも容易な制御回路及びメモリコントローラを提供することができる。

##### 【図面の簡単な説明】



【図 1】 リードソロモン符号方式の概要を示す図。

【図 2】 本発明による制御回路の第 1 の実施の形態の構成を示すブロック図。

【図 3】 本発明による制御回路の第 2 の実施の形態の構成を示すブロック図。

【図 4】 本発明による制御回路の第 1、第 2 の実施の形態に適用される動作テストを示すフローチャート。

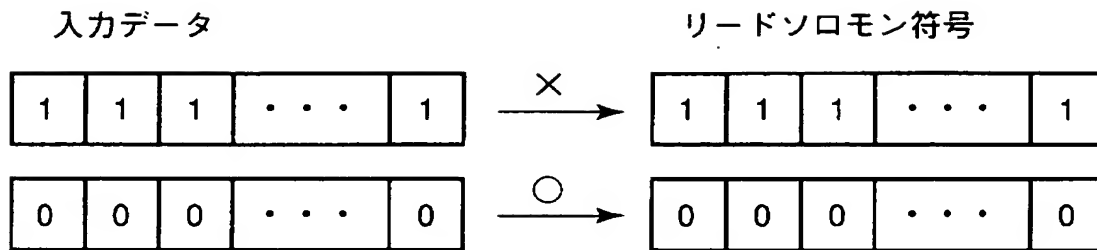
【図 5】 図 4 に示すフローチャートの機能シミュレーションの詳細を示すフローチャート。

【符号の説明】

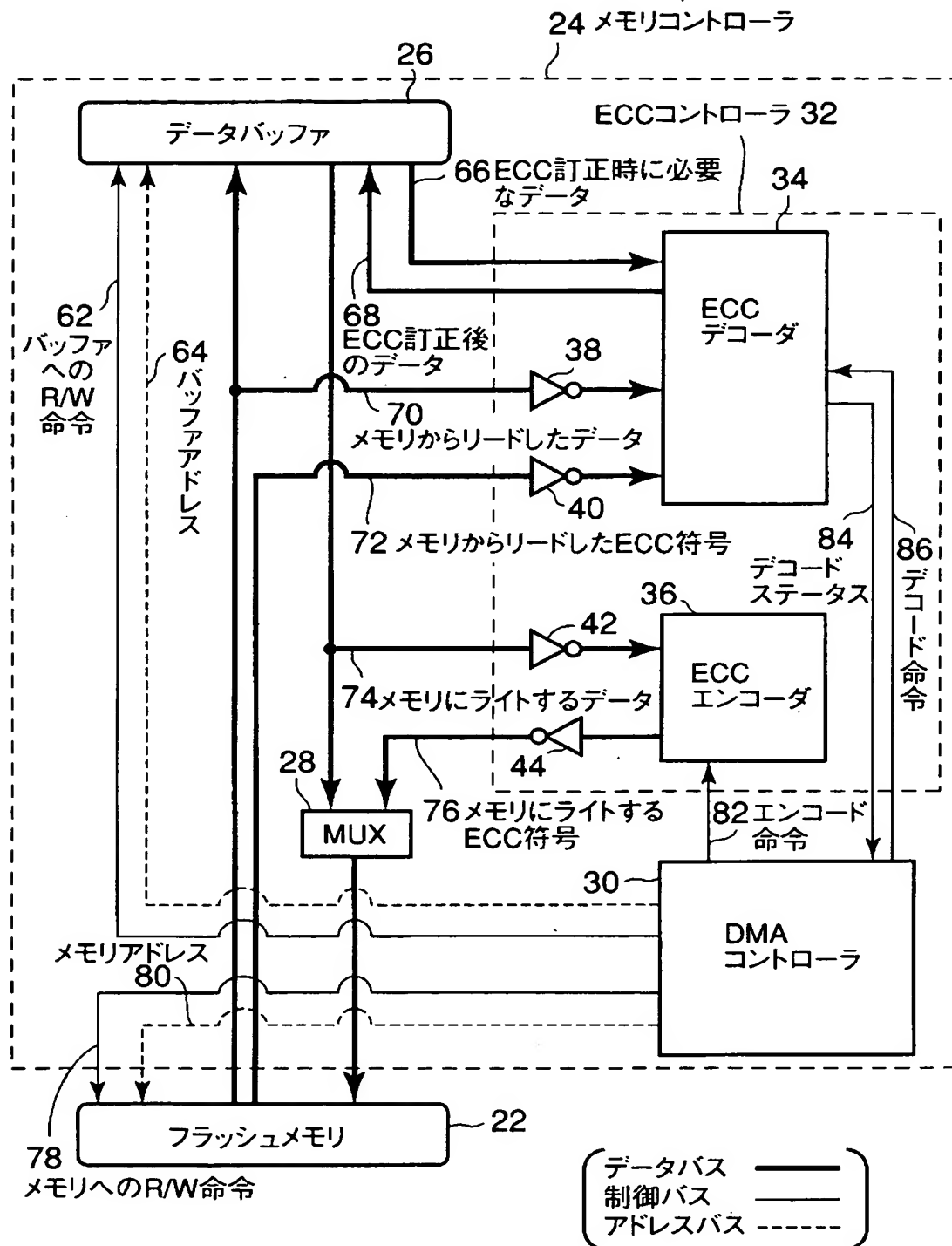
2 2…フラッシュメモリ、2 4…メモリコントローラ、2 6…データバッファ、2 8…マルチプレクサ、3 0…DMAコントローラ、3 2…ECCコントローラ、3 4…ECCデコーダ、3 6…ECCエンコーダ、3 8、4 0、4 2、4 4…インバータ

【書類名】 図面

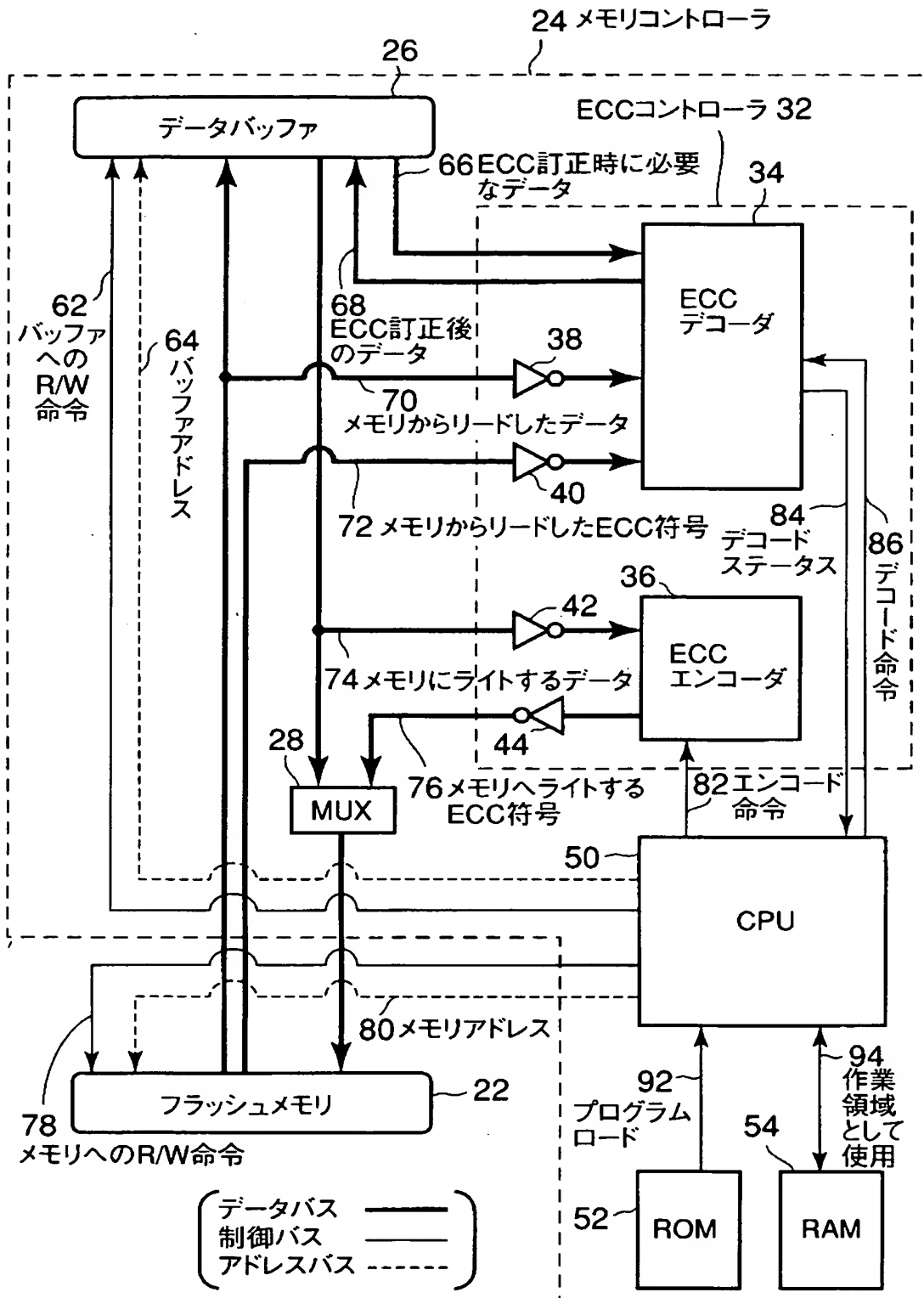
【図 1】



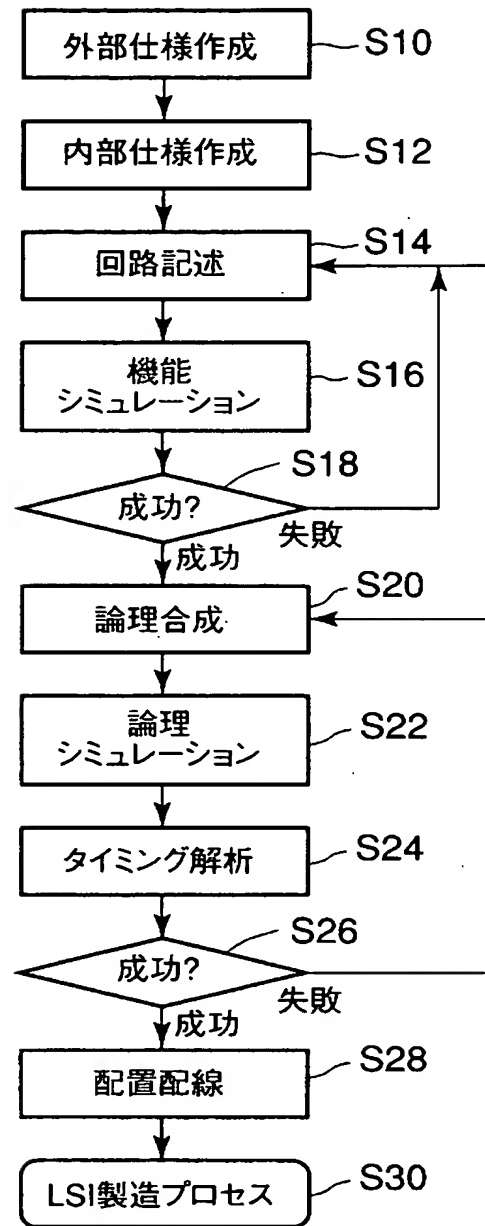
【図 2】



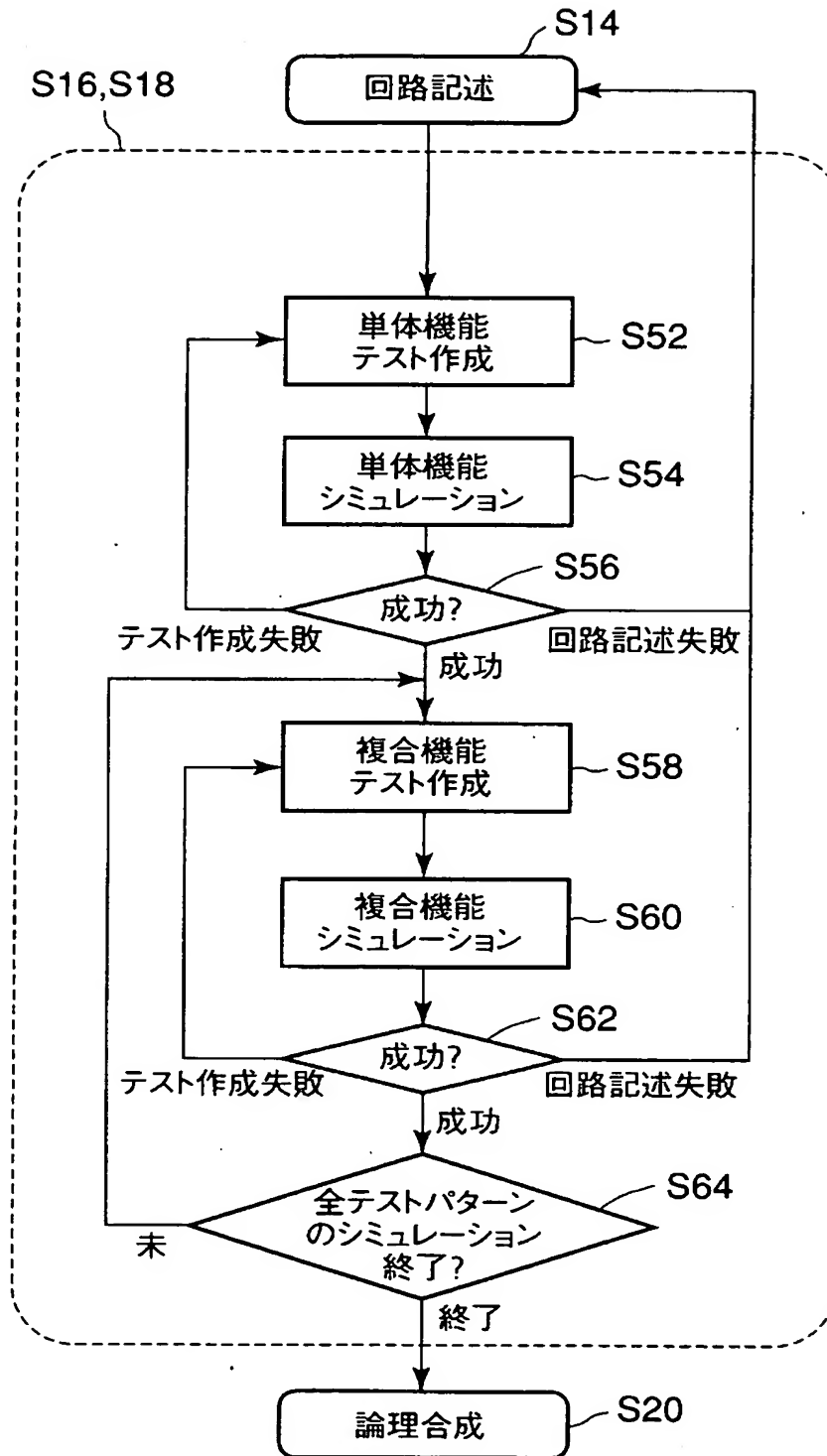
【図 3】



【図 4】



【図 5】



【書類名】 要約書

【要約】

【課題】 消去後の初期値に対して E C C エラーを誤検出することがなく、動作テストも容易な制御回路及びメモリコントローラを提供すること。

【解決手段】 フラッシュメモリ（2 2）に E C C 符号として、データの反転値から計算されたリードソロモン符号の反転値が書き込まれる。データのビット列がオール“1”の場合は、オール“1”の E C C 符号が書き込まれる。フラッシュメモリ（2 2）から消去後の初期値（オール“1”）を読み出す場合、E C C デコーダ（3 4）に入力されるデータのビット列はオール“0”、E C C 符号もオール“0”となるため、E C C エラーが検出されることがない。

【選択図】 図 2

特願 2003-024861

出願人履歴情報

識別番号

[000003078]

1. 変更年月日            2001年   7月   2日  
   [変更理由]            住所変更  
                            東京都港区芝浦一丁目1番1号  
                            株式会社東芝
  
2. 変更年月日            2003年   5月   9日  
   [変更理由]            名称変更  
                            住所変更  
                            東京都港区芝浦一丁目1番1号  
                            株式会社東芝